# Labeling Points on a Single Line*

Yu-Shin Chen, D.T. Lee†, and Chung-Shou Liao‡
Institute of Information Science
Academia Sinica Nankang, Taipei, Taiwan
Email: puc@cht.com.tw
dtlee@iis.sinica.edu.tw
shou794@iis.sinica.edu.tw

November 30, 2004

**Abstract**

In this paper, we consider a map labeling problem where the points to be labeled are restricted on a line. It is known that the 1d-4P and the 1d-4S unit-square label placement problem and the Slope-4P unit-square label placement problem can both be solved in linear time and the Slope-4S unit-square label placement problem can be solved in quadratic time in [7]. We extend the result to the following label placement problem: Slope-4P unit-height (width) label placement problem and elastic labels and present a linear time algorithm for it provided that the input points are given sorted. We further show that if the points are not sorted, the label placement problems have a lower bound of $\Omega(n \log n)$, where $n$ is the input size, under the algebraic computation tree model. Optimization versions of these point labeling problems are also considered.

## 1 Introduction

When drawing a map, in order to let people know what is on the map, the main approach is attaching texts or labels to geographic features on the map. How to place labels so that they do not overlap, is a well-known important problem in cartography. Thus, there are

*Supported in part by the National Science Council under Grants NSC-92-2213-E-001-024, NSC-93-2213-E-001-013, NSC-93-2422-H-001-0001, and NSC-93-2752-E-002-005-PAE.

†Also with the Dept. of Computer Science and Information Engineering, National Taiwan University. Email:dtlee@csie.ntu.edu.tw

‡Also with the Dept. of Computer Science and Information Engineering, National Taiwan University.

many research results on this topic and many algorithms have been developed for labeling points that are on lines [7, 8, 9, 15] or in a region [5, 6, 10, 11, 12, 14, 17, 19]. In the ACM Computational Geometry Impact Task Force report [2] the map label placement is listed as an important research area.

Let $S$ denote a set of points $S = \{p_1, p_2, \ldots, p_n\}$ in the plane, called *anchors*. Associated with each anchor there is an axis-parallel rectangle, called *label*. The *point-feature label placement problem* or simply *point labeling problem*, is to determine a placement of these labels such that the anchors coincide with one of the corners of their associated labels and no two labels overlap. The point labeling problem for labeling an arbitrary set of points has been shown to be NP-complete [6, 10, 11, 14], and some heuristic algorithms were presented in [4, 6, 19].

There are many variations of the point labeling problem, including shapes of the labels, locations of the anchors to be labeled and where the labels are placed. Three common shapes of the labels considered were circles, squares and rectangles. Sometimes, the rectangular labels may be given additional constraints, e.g., the width and height may satisfy a certain aspect ratio. The placement of the labels may be restricted. For instance, the anchors must be on the corners of the labels (i.e., fixed-position model), or on the boundary of the label (i.e., sliding model). One may also consider an optimization version of the problem, e.g., maximizing the size of labels.

Strijk and Wolff [17] showed that to decide if a set of points can be labeled with unit circles is NP-hard, and also that there is a constant $0 < \delta < 1$ such that it is NP-hard to label points with uniform circles of diameter greater than $\delta \cdot d_{opt}$, where $d_{opt}$ denotes the optimal diameter. They presented an approximation algorithm for labeling points with uniform circles whose diameter is about $1/19.59$ times the optimal in $O(n \log n)$ time [17].

The *elastic point labeling problem*, is yet another variation, and is defined as follows: Given a set of $n$ points on the plane, each of which is associated with an *elastic* rectangular label of a given area, we want to choose a valid height (or width) for each label and a corner to place at its associated anchor so that no two labels overlap. A label is said to be *elastic* if its width and height can vary but its area remains a constant.

Iturriaga and Lubiw [8] proved that the *one-corner elastic point labeling problem*, in which the anchors must be at the corners of the labels is NP-hard. They considered the elastic point labeling problem when $n$ anchors lie on the $x$-axis and $m$ anchors lie on the $y$-axis, and the labels for the anchors on the $x$-axis (resp. $y$-axis) have an edge coincident with the $x$-axis (resp. $y$-axis) lying in the same quadrant [8] and presented an $O(nm)$ algorithm. They also considered the *rectangle perimeter point labeling problem* [9], in which the given elastic labels are to be placed on anchors that lie on the boundary of a given rectangular map. They combined the solution of the two-axis case and the two-parallel-line case to solve the rectangle perimeter point labeling problem in $O(n^4)$ time, where $n$ is the number of anchors. The two-parallel-line case is similar to the two-axis case except that the points to be labeled lie on two parallel lines.

In this paper we consider the case when the anchors lie on a line and are to be labeled with rectangular labels. We consider two main models. One is *fixed-position model*, denoted 4P model, in which a label must be placed so that the anchor coincides with one of its four corners, and the other is *sliding model*, denoted 4S model, in which a label can be placed so that the anchor lies on one of the four boundary edges of the label. Figure 1 shows these two point labeling models, among others, that have been studied previously [12, 15]. The positions {1, 2, 3, 4} in 4P model shown in Figure 1 denote the corner positions of labels coincident with the anchor, and the arrows in 4S model indicate the directions along which the label can slide, maintaining contact with the anchor.

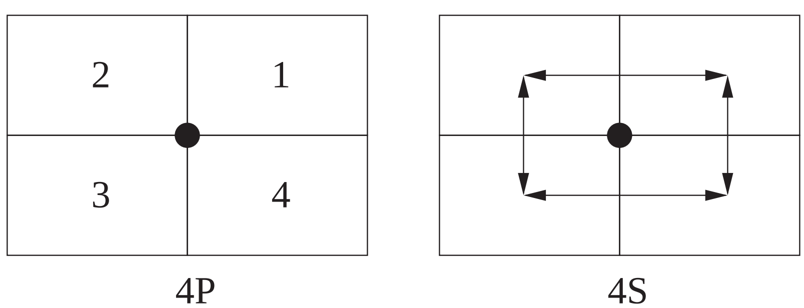


Figure 1: Illustration of 4P and 4S models.

Extending the result of [7], we consider the problem of labeling points on a sloping line with rectangular labels of unit-height (or unit-width) in 4P model, referred to as *Slope-4P unit-height label placement problem* and the problem of maximizing the size of the rectangular equal-width labels of the points on a horizontal line whose top edge or bottom edge coincide with the line in 4S model, referred to as *1d-4S equal-width label maximization problem*. We also address the elastic point labeling problem, where points to be labeled lie on a sloping line.

This paper is organized as follows. In Section 2, we give some definitions and notation about our point labeling problems. In Section 3, we consider the Slope-4P unit-height label placement problem for points lying on a sloping line. In Section 4, we extend the result of Section 3 to the case where the labels are elastic. In Section 5, we consider the 1d-4S equal-width label maximization problem, and give a lower bound of the time complexity of the 1d-4S equal-width label placement (decision) problem.

# 2 Preliminaries

In this section, we introduce some terminology and notation. Consider a set $P = \{p_1, \ldots, p_n\}$ of $n$ anchors on a line $L$ such that they are sorted in ascending $x$-coordinate, i.e., $p_i.x < p_{i+1}.x$, $i = 1, 2, \ldots, n-1$, and for each $p_i$, there is an axis-parallel rectangular label $l_i$.

We shall use prefixes, 1d- or Slope- to refer to the problems in which the anchors lie on a horizontal or a sloping line, respectively. Combined with 4P and 4S model as described previously, we have, for example, 1d-4P denotes that the anchors lie on a horizontal line and each label has four possible placement positions, and Slope-4S denotes that the anchors lie on a sloping line and each anchor can lie on any of the four boundary edges of a label.

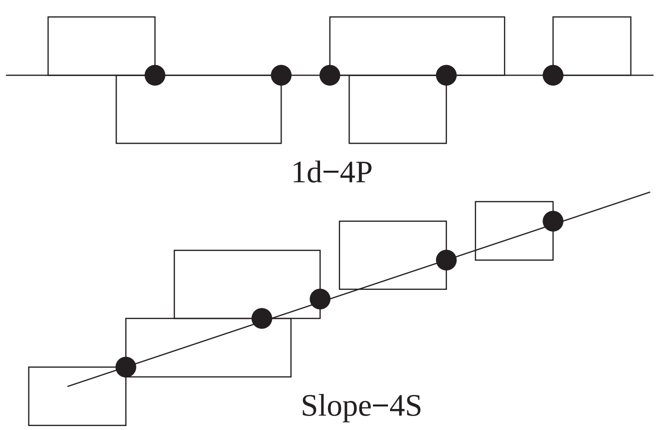


Figure 2: An illustration of a good 5-realization in 1d-4P and Slope-4S models.

A $k$-tuple $R_k = (r_1, \cdots, r_k)$ is a $k$-realization of labels $l_1, l_2, \ldots, l_k$ for $P$ such that each $r_i$ encodes a position of $l_i$. If no two labels in $R_k$ intersect each other, $R_k$ is said to be a *good $k$-realization*. Our goal is to find a good $n$-realization $R_n$, or good realization for short, for a given $P$. See, for example, Figure 2 for a good 5-realization in 1d-4P and Slope-4S models.

A point $u$ is said to be *above* another point $v$, or $v$ is said to be *below* $u$ if the $y$-coordinate of $v$, $v.y$, is smaller than that of $u$, $u.y$, i.e., $v.y < u.y$ . A point $u$ is said to be *below*, *on* and *above* a line $L$, if the $y$-coordinate of $u$ is smaller than, equal to, and greater than, respectively, the $y$-coordinate of the vertical projection point $u_L$ of $u$ on $L$. Let $\delta$ denote the length of the unit for the unit-height label placement problem discussed in this paper such that each anchor has a rectangular label of the same height $\delta > 0$.

# 3 Labeling Points on a Sloping Line with Unit-Height Labels

Garrido et al. solved the Slope-4P unit-square label placement problem in $O(n)$ time, and the Slope-4S unit-square label placement problem in $O(n^2)$ time [7]. They made use of a concept, called *shadow*, which is determined by the last two labels. For our Slope-4P unit-height label placement problem this concept unfortunately does not work. Figure 3 shows that the *shadow* of the realization $R = \{r_1, r_2, r_3, r_4\}$ defined by the last two labels as stated in Lemma 4 in [7] will produce an incorrect labeling for the next label at $p_5$. We therefore propose a new idea, called *top* and *bottom domination labels*, for the Slope-4P unit-height label placement problem. It is obvious that the case for unit-width labels is similar, so we consider only the case for unit-height labels. We assume that the slope of the line is positive without loss of generality.

**Definition 3.1** *In a realization $R$, a label $l_i$ is called a top label, if its upper-right corner $q_i$ is above the sloping line, and a bottom label, if $q_i$ is below or on the sloping line.*

**Definition 3.2** *In a good realization $R$, the feasible region, denoted $F(l_i)$, of each label $l_i$ with respect to label $l_j$, for all $j > i$, or simply the feasible region of each label $l_i$, is defined as the possible region the label $l_j$, $j > i$, can be placed without intersecting $l_i$. Specifically,*

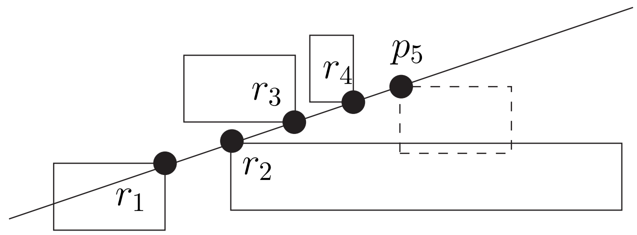


Figure 3: An illustration of how the notion of shadow of the realization $R = \{r_1, r_2, r_3, r_4\}$ as stated in [7] is not adequate to determine the placement of the fifth label.

*if $l_i$ is a bottom label, $F(l_i)$ is the intersection of $ND_{q_i}$ and $A_{b_i}$, where $ND_{q_i}$ is the locus of points not dominated* [1] *by the upper-right corner $q_i$ of $l_i$ and $A_{b_i}$ is the region containing points whose y-coordinates are no less than the y-coordinate of the bottom boundary edge of $l_i$; if $l_i$ is a top label, $F(l_i)$ is the intersection of $ND_{q_i}$ and $A_{t_i}$ , where $ND_{q_i}$ is defined as above, and $A_{t_i}$ is the region containing points whose y-coordinates are no less than the y-coordinate minus $\delta$ of the bottom boundary edge of $l_i$.*

**Definition 3.3** *In a good realization $R$, label $l_j$ is said to cover label $l_i$ or label $l_i$ is covered by label $l_j$, for $j > i$, if $F(l_j) \subseteq F(l_i)$. That is, whenever label $l_k$ can be placed in $F(l_j)$ it can be placed in $F(l_i)$ for $i < j < k$ (see Figure 4).*

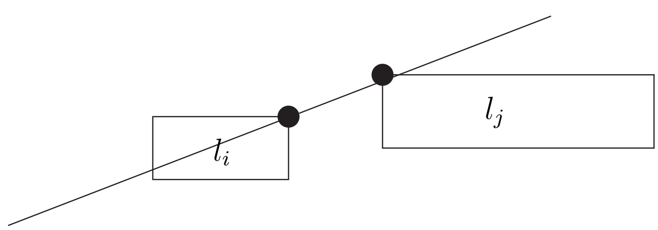


Figure 4: Label $l_j$ covers $l_i$

**Definition 3.4** *In a good realization $R$, the last top label, is called the top domination label, denoted $d_t$, and the last bottom label, the bottom domination label, denoted $d_b$.*

**Lemma 3.5** *The pair of labels $d_t$ and $d_b$ covers all other labels in any good realization.*

[1]A point $u$ is said to *dominate* another point $v$, or $v$ is said to *be dominated* by $u$, if both $x$- and $y$-coordinates of $v$ are no greater than those of $u$.

**Proof:** For an arbitrary good realization $R$, the feasible region $F(l_i)$ of each top label $l_i$ contains the feasible region $F(d_t)$ of $d_t$ obviously. So $d_t$ covers all the top labels. Similarly, since $F(d_b) \subseteq F(l_j)$ for each bottom label $l_j$, $d_b$ covers all the bottom labels. The lemma follows. $\square$

Let the top and bottom domination labels associated with a good $k$-realization $R_k$ be denoted $d_{t_k}$ and $d_{b_k}$ respectively.

**Definition 3.6** *For a good $k$-realization $R_k$, we define its associated* `envelope region` *$E_k$ as the intersection of $F(d_{b_k})$ and $F(d_{t_k})$ (see Figure 5). Given two $k$-realizations, $R^1_k$, and $R^2_k$, we say that the domination pair $d^1_{b_k}$ and $d^1_{t_k}$ associated with $R^1_k$, covers the domination pair $d^2_{b_k}$ and $d^2_{t_k}$ associated with $R^2_k$, if the envelope region $E^2_k$ contains the envelope region $E^1_k$, i.e., $E^1_k \subseteq E^2_k$. Two $k$-realizations $R^1_k$, and $R^2_k$ are said to be comparable if $E^1_k \subseteq E^2_k$ or if $E^2_k \subseteq E^1_k$. Otherwise, they are said to be incomparable.*

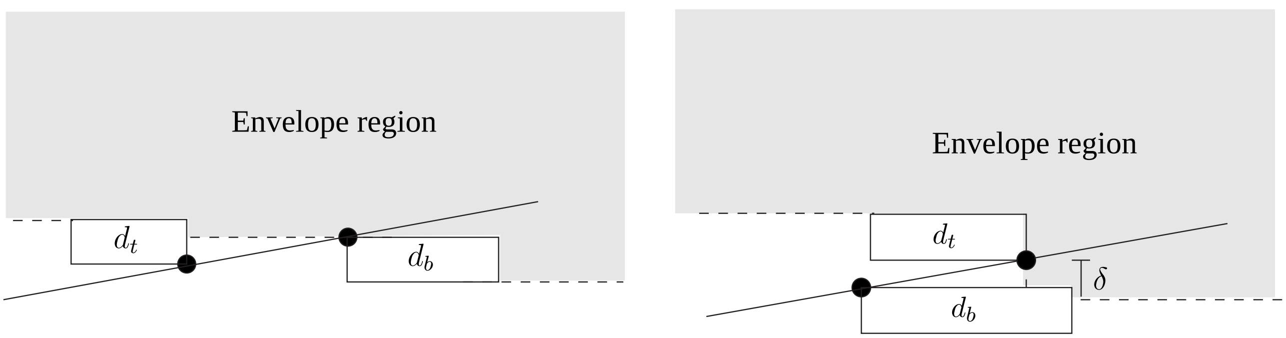


Figure 5: The envelope region.

Lemma 3.5 plays an important role for Slope-4P model and it implies that if a new label $l_{k+1}$ is added to a $k$-realization $R_k$, we only need to check if label $l_{k+1}$ can be placed to lie totally in the envelope region associated with $R_k$, and if so, which positions $l_{k+1}$ can be placed so as to obtain a good $(k+1)$-realization $R_{k+1}$.

Therefore, we will simply use the ordered pair $(d_{b_k}, d_{t_k})$ of bottom and top domination labels of $R_k$ to represent the $k$-realization, as it defines the envelope region $E_k$ associated with $R_k$.

For convenience, we shall encode the label positions in a $k$-realization, using elements in $\{1, 2, 3, 4\}$, as described in Figure 1. That is, $R_k = \{r_1, r_2, \ldots, r_k\}$ is a $k$-realization, where $r_i \in \{1, 2, 3, 4\}$, for $1 \le i \le k$.

It turns out that given a $k$-realization $R_k$, there are *six* cases of possible positions that $l_{k+1}$ can be placed as follows.

1. $r_{k+1} \in \{1, 2, 3, 4\}$.
2. $r_{k+1} \in \{1, 2, 4\}$.
3. $r_{k+1} \in \{1, 4\}$.
4. $r_{k+1} \in \{1, 2\}$.
5. $r_{k+1} \in \{1\}$.
6. $r_{k+1} = \emptyset$.

And the notion of envelope given in Definition 3.6 simplifies our algorithm. We shall adopt an incremental algorithm by placing labels one at a time in ascending order of the $x$-coordinates of the anchors on the sloping line. The algorithm is *greedy* in nature. That is, when we consider adding the next label, $l_{k+1}$ to a $k$-realization $R_k$, if we have more than one choice to put the label, we shall place it so that the resulting envelope associated with the new $(k+1)$-realization is as *maximal* as possible, as will be shown below. Thus, we adopt the following *maximal placement* strategy.

1. If $r_{k+1} \in \{1, 2, 3, 4\}$, we let $r_{k+1} = 3$.
2. If $r_{k+1} \in \{1, 2, 4\}$, we let $r_{k+1} = 2$ or $r_{k+1} = 4$, as the resulting realizations may be *incomparable*.
3. If $r_{k+1} \in \{1, 4\}$, we let $r_{k+1} = 4$.
4. If $r_{k+1} \in \{1, 2\}$, we let $r_{k+1} = 2$.
5. $r_{k+1} \in \{1\}$.
6. $r_{k+1} = \emptyset$.

**Lemma 3.7** *For anchors $\{p_1, p_2, \ldots, p_i\}$, there are at most two incomparable $i$-realizations. for $1 \le i \le n$.*

**Proof.** We prove it by induction on label index $i$. For $i = 1$ and $i = 2$, this lemma holds trivially. Suppose that the induction hypothesis holds for $i = k$. Consider $i = k + 1$. By induction hypothesis, there are at most two $k$-realizations $R_k^1$ and $R_k^2$. Let $(d_{b_k}^j, d_{t_k}^j)$ represent the bottom and top domination labels associated with $R_k^j$ respectively, $j = 1, 2$. For each of the two $k$-realization $R_k$, there are at most two feasible position choices for $r_{k+1}$ at the same time according to the maximal placement strategy mentioned above.

Note that $r_{k+1}$ can be either $d_{b_{k+1}}$ or $d_{t_{k+1}}$ in $R_{k+1}$. Thus, the following are the four $(k+1)$-realizations: $(r_{k+1}, d_{t_k}^1)$ represents $R_{k+1}^{1b}$, $(d_{b_k}^1, r_{k+1})$ represents $R_{k+1}^{1t}$, $(r_{k+1}, d_{t_k}^2)$ represents $R_{k+1}^{2b}$ and $(d_{b_k}^2, r_{k+1})$ represents $R_{k+1}^{2t}$. We distinguish the following cases to show that there are actually at most two incomparable $(k+1)$-realizations.

**Case 1:** Assume $r_{k+1} = 3$ (when $r_{k+1} \in \{1, 2, 3, 4\}$) for one of $R_k^1$ and $R_k^2$, say $R_k^1$ without loss of generality. In this case, we have at most three possible realizations: $R_{k+1}^{1b}$, $R_{k+1}^{2b}$, and $R_{k+1}^{2t}$. Since $E_{k+1}^{2t} \subseteq E_{k+1}^{1b}$, we have at most two $(k+1)$-realizations $R_{k+1}^{1b}$ and $R_{k+1}^{2b}$.

**Case 2:** Assume $r_{k+1} = 4$ (when $r_{k+1} \in \{1, 4\}$) for one of $R_k^1$ and $R_k^2$, say $R_k^1$ without loss of generality. There are three realizations: $R_{k+1}^{1b}$, $R_{k+1}^{2b}$, and $R_{k+1}^{2t}$.

**Subcase 2-1:** If $r_{k+1} = 2$ or 4, that is, $r_{k+1} \in \{1, 2, 4\}$ for $R_k^2$, then obviously $R_{k+1}^{1b}$ and $R_{k+1}^{2b}$ are comparable, and one can be eliminated.

**Subcase 2-2:** If $r_{k+1} \in \{1, 4\}$ for $R_k^2$, no $R_{k+1}^{2t}$ exists. (The subcase $r_{k+1} \in \{1, 2\}$ is similar.)

**Subcase 2-3:** If $r_{k+1} = 1$ for $R_k^2$, then either $E_{k+1}^{2b} \subseteq E_{k+1}^{1b}$ (as $r_{k+1} = d_b$) or $E_{k+1}^{2t} \subseteq E_{k+1}^{1b}$ (as $r_{k+1} = d_t$) and hence one can be eliminated.

**Case 3:** Assume $r_{k+1} = 2$ (when $r_{k+1} \in \{1, 2\}$) for one of $R_k^1$ and $R_k^2$, say $R_k^1$ without loss of generality. There are three realizations: $R_{k+1}^{1t}$, $R_{k+1}^{2b}$, and $R_{k+1}^{2t}$.

**Subcase 3-1:** If $r_{k+1} = 2$ or $r_{k+1} = 4$, that is, $r_{k+1} \in \{1, 2, 4\}$ for $R_k^2$, then since $r_{k+1} = 4$ is feasible for $R_k^2$, $E_{k+1}^{1t} \subseteq E_{k+1}^{2t}$.

**Subcase 3-2:** If $r_{k+1} \in \{1, 2\}$ for $R_k^2$, no $R_{k+1}^{2b}$ exists. (The subcase $r_{k+1} = 1$ is similar.)

**Case 4:** Assume $r_{k+1} = 1$ for one of $R_k^1$ and $R_k^2$, say $R_k^1$ without loss of generality. There are three realizations: $R_{k+1}^{1b}$, $R_{k+1}^{2b}$, and $R_{k+1}^{2t}$, if $r_{k+1} = d_b$, and $R_{k+1}^{1t}$, $R_{k+1}^{2b}$, and $R_{k+1}^{2t}$, if $r_{k+1} = d_t$.

**Subcase 4-1:** If $r_{k+1} = 2$ or $r_{k+1} = 4$, that is, $r_{k+1} \in \{1, 2, 4\}$ for $R_k^2$, then $E_{k+1}^{1b} \subseteq E_{k+1}^{2b}, E_{k+1}^{2t}$ if $r_{k+1} = d_b$. (As $r_{k+1} = d_t$, it's similar.)

**Subcase 4-2:** If $r_{k+1} = 1$ for $R_k^2$, no $R_{k+1}^{2t}$ exists as $r_{k+1} = d_b$. (As $r_{k+1} = d_t$, it's similar.)

**Case 5:** Assume $r_{k+1} \in \{1, 2, 4\}$ for $R_k^1$ and $r_{k+1} \in \{1, 2, 4\}$ for $R_k^2$, there are four realizations: $R_{k+1}^{1b}$, $R_{k+1}^{1t}$, $R_{k+1}^{2b}$, and $R_{k+1}^{2t}$. Since $r_{k+1} = 4$ is feasible for both $R_k^1$ and $R_k^2$, so $R_{k+1}^{1t} = R_{k+1}^{2t}$. For both $R_{k+1}^{1b}$ and $R_{k+1}^{2b}$, $r_{k+1} = 4$. So they are comparable and one can be eliminated.

**Case 6:** Assume $r_{k+1} = \emptyset$ for one of $R_k^1$ and $R_k^2$, it's trivial.

For every case, there are at most two incomparable $(k+1)$-realizations. $\square$

We thus have immediately the following algorithm, Algorithm Slope-4P, for finding a good realization for the Slope-4P unit-height label placement problem, using an iterative method by considering the labels one at a time in order.

**Algorithm Slope-4P.**

**Input:** A sloping line $L$ with positive slope, a set of anchors $P = \{p_1, p_2, \ldots, p_n\}$ on $L$ sorted in ascending $x$-coordinate, and the corresponding label width $l_1, l_2, \ldots, l_n$ (the heights of all the labels are identical, denoted $\delta$ ).

**Output:** A good realization $R$ of $P$ if it exists, and *nil* otherwise.

**Method.**

1. Assume that we maintain at most two realizations $R^1, R^2$ for the set $P$ of anchors and let their associated bottom and top domination labels be $(d_b^1, d_t^1)$, $(d_b^2, d_t^2)$ respectively.

2. Let $P^1, P^2$ denote the set ($\subseteq \{1, 2, 3, 4\}$) of positions the next (new) label can be placed with respect to $R^1, R^2$ respectively;

3. We consider the anchors in ascending order of their $x$-coordinates one at a time;

4. $R^1, R^2, P^1, P^2 \leftarrow \emptyset$ and $d_b^1, d_t^1, d_b^2, d_t^2 \leftarrow nil$;

5. For each realization $R^j, j = 1, 2$, we keep two temporary realizations, $R^{j\_b}$, and $R^{j\_t}$;

6. **For** $i = 1$ **to** $n$

   Determine $P^j$ with respect to $(d_b^j, d_t^j)$, for $r_i$, $j = 1, 2$;

   **If** ($P^1 = \emptyset$ and $P^2 = \emptyset$) **then return** *nil* and break;

   Temporarily store possible realizations in $R^{j\_b}$, $R^{j\_t}$ for $P^j, j = 1, 2$, according to the **maximal placement strategy**;

   Compare $R^{1\_b}$, $R^{1\_t}$, $R^{2\_b}$, $R^{2\_t}$ to get at most two maximal realizations;

   Update $R^j$ from the previous step, and also update $(d_b^j, d_t^j)$, for $j = 1, 2$;

   **END**;

**Theorem 3.8** *The* `Slope-4P` *unit-height (width) label placement problem can be solved in linear time when the input anchors are given sorted.*

**Proof.** By Lemma 3.7 there are at most two incomparable realizations that need to be maintained at each step, and the envelope regions of these two realizations are maximal. Thus, the algorithm Slope-4P is able to find at least one good realization for the input anchor set $P$, if there exist feasible solutions.

As we determine the possible label positions of the next label, only two realizations ($d_b^1$, $d_t^1$), ($d_b^2$, $d_t^2$) of bottom and top domination labels have to be checked according to previous lemma. It takes constant time in each iteration. Therefore the total time complexity of Algorithm Slope-4P is $O(n)$. □

# 4 Labeling Points on a Sloping Line with Elastic Labels

We now consider the case when the labels are elastic. An *elastic rectangle* $E$ is a family of rectangles specified by a triplet $(\alpha, H, W)$, where $\alpha$ is the area of any rectangle in $E$, $H = [h^{min}, h^{max}]$ is the range of the height of the rectangles, $W = [w^{min}, w^{max}]$ is the range of the width. Given an anchor $p$, the placement of an elastic label $E$ will be specified by $(p, Q)(E)$, where $Q \subseteq \{1, 2, 3, 4\}$ is a set of possible label position indexes and the anchor $p$ is to coincide with a corner of a rectangle in $E$. The position index is the same as in the 4P model defined previously. Note that given a label position $p(E)$, the family $E$ of rectangles is described by a hyperbolic segment tracing out the locus of the corner of the elastic rectangle opposite to $p$. In our model, we assume $h^{min}$ and $w^{min}$ are both set to be unity. Figure 6 shows an elastic rectangle with $Q(E) = \{2\}$, and the hyperbola shown as a dashed curve. The notation $r_{min}$ denotes the **unit rectangle** of height $h^{min}$ and width $w^{min}$.

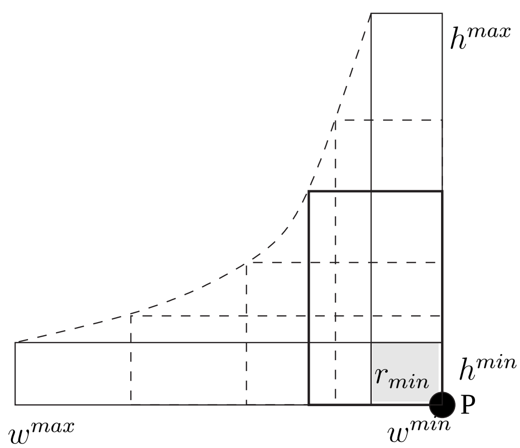


Figure 6: An instance of an elastic rectangle

**Theorem 4.1** *For a set of anchors, the* `Slope-4P` *elastic label model has a good realization if and only if* `Slope-4P` *unit-width label model (slope is less than or equal to* $\frac{h^{min}}{w^{min}}$ *or* `Slope-4P` *unit-height label model (slope is greater than or equal to* $\frac{h^{min}}{w^{min}}$ *) has a good realization.*

**Proof.** If there exists a good realization for Slope-4P unit-width (respectively height) label placement problem instance, then this solution is a solution to the Slope-4P elastic label placement problem in which width $w = \delta$ (respectively height $h = \delta$), and height $h = \alpha/\delta$ (respectively width $w = \alpha/\delta$). Conversely, suppose there exists a good realization $R_e$ for Slope-4P elastic label placement problem instance. Let us consider the case when the slope of the sloping line is less than or equal to $\frac{h^{min}}{w^{min}}$ and the other case when the slope is greater than or equal to $\frac{h^{min}}{w^{min}}$ is similar. We claim that we can let all elastic labels be unit-width labels, where the unit width $\delta$ is set to be $w^{min}$ and obtain a good realization. First, it is obvious that there exists a good realization $R'$ for Slope-4P unit-rectangle $r_{min}$ label model for the input instance by reducing each label in $R_e$ to $r_{min}$. Next, we let each top label in $R'$ *grow* upwards to be of height $h = \alpha/w^{min}$. Similarly, we let each bottom label in $R'$ *grow* downward to be of height $h = \alpha/w^{min}$. Since the slope of the sloping line is less than or equal to $\frac{h^{min}}{w^{min}}$, there is no intersection for every label of width $w = w^{min}$ and height $h = \alpha/w^{min}$. Then we have a good realization for Slope-4P unit-width label placement problem, where $\delta = w^{min}$. □

# 5 Maximizing Label Sizes and the Lower Bound

## 5.1 Maximizing Label Size for Slope-4P Unit-height Model

The maximization version of Slope-4P unit-height label model (Slope-4P unit-width label model, similarly) is defined as follows: Given a set of $n$ anchors $P = \{p_1(x_1, y_1), p_2(x_2, y_2), \ldots, p_n(x_n, y_n)\}$ on a sloping line with positive slope, where $(x_i, y_i)$ are the $x$- and $y$-coordinates of anchor $p_i, i = 1, 2, \ldots, n$, and each anchor $p_i$ is associated with a rectangular label of unit height $h = \delta$ and width $l_i$, find the maximum stretch factor $\lambda_{max}$ such that the width and height of each label are multiplied by $\lambda_{max}$ for which there exists a good realization. For this maximization problem, we can use an idea similar to that used in [7] to solve this problem and the following result immediately follows.

**Theorem 5.1** *The maximization version of* `Slope-4P` *unit-height (width) label placement problem can be solved in time $O(n^2 \log n)$.*

## 5.2 Maximizing Label Size for 1d-4S Equal-width Model

The maximization version of 1d-4S equal-width label placement problem is defined as follows: given a set $P = \{p_1, p_2, \ldots, p_n\}$ of $n$ anchors on a horizontal line, find the maximum label width $w^{max}$ such that there is a good realization of $P$. We show below that this problem can be solved in $O(n^2)$ time instead. We use an iterative approach to deal with this problem. Following Theorem 4 in [7], we modify their dynamic programming method and obtain an alternating good realization in a greedy manner [3] for the 1d-4S equal-width label placement problem in linear time, assuming the input anchors are given sorted. That is, we place labels as left as possible for each anchor, and alternately place labels above and below the line in sorted order. [2] Therefore, we can separate the point set $P$ into two parts, $P_o = \{p_i \mid i \text{ is odd }\}$ and $P_e = \{p_i \mid i \text{ is even }\}$. The labels of $P_o$ are placed above the line and the labels of $P_e$ are placed below the line. The solutions $w_o^{max}$ and $w_e^{max}$ of $P_o$ and $P_e$, respectively can be computed independently and the smaller of $w_o^{max}$ and $w_e^{max}$ is the solution. We shall consider only the problem of $P_o$ and the other problem is similar. This problem is equivalent to finding an interval of maximum length for each anchor in $P_o$ such that each interval is of equal length, and there is **exactly** one anchor in each interval. In other words, we would like to place cuts on the horizontal line to form a collection of equal-length cut intervals, each containing exactly one anchor.

We would proceed by processing these anchors in $P_o$ one at a time from left to right and placing cuts in an appropriate manner. Assume that these anchors are ordered from left to right as $p_i$, $i = 1, 2, \ldots, m$. Let $x_i$ denote the $x$-coordinate of the $i$th anchor and $c_i$ denote the $x$-coordinate of $i$th cut. We use $(x_m - x_j)/(m - j - 1)$ to be the (weighted) cut interval length, denoted $X$. Initially $j = 1$, clearly the first cut can be placed at the leftmost point, thus $c_1 = x_1$. Note that $X$ is an upper bound on the optimal label width. We then proceed to place cut $c_i$ which is at distance $X$ to the right of cut $c_{i-1}$ for $i \geq 2$, until either there is more than one anchor in the cut interval or there is no anchor in the cut interval. When either case occurs, we need to adjust the cut interval length $X$. For each anchor $p_i$, we maintain a safety clearance $sc_i = |c_i - x_i|$, weighted clearance $wc_j = sc_j/(j-1)$, $j \geq 2$, and an invariant condition that there is exactly one anchor in each cut interval.

**Case 1:** Suppose there is no anchor in the cut interval or cut $i$ happens to be placed at the $i$th anchor, i.e. $c_i = x_i$.

In this case we can separate the rest of anchors from what we have processed, and treat the remaining set of anchors as a new subproblem. In the former case we will search for the next anchor $p_j$, rename it $p_1$, place a cut $c_1$ at it and update $X = (x_m - x_j)/(m-j-1)$.

[2] It is shown in [3] that one can always obtain an alternate label placement, if there is a feasible solution for the 1d-4S equal-width label placement problem

While in the latter case, anchor $p_i$ would play the role of $p_1$ and cut $c_i$ would play the role of $c_1$, and this process continues.

**Case 2:** Suppose there is more than one anchor in the cut interval.

Assume that this happens for cut $c_{k+1}$ and that the distance between cut $c_k$ and the second anchor $p_{k+2}$ to its right is $Y$, $Y < X$. This means that the label width $X$ is too large and we need to re-adjust the length of cut interval. We first calculate the new cut interval length $X' = [(k-1)X + Y]/k$ which is strictly less than $X$. Let $D = X - X'$. Let $U$ be the minimum weighted clearance, i.e., $U = min\{wc_j \mid 2 \leq j \leq k\}$. If $D < U$, then we have a new arrangement of cuts and the invariant condition that each cut interval has exactly one point is maintained. We then update the weighted clearances for all the anchors by decrementing each $wc_j$ by $D$, place cut $c_{k+1}$ at anchor $p_{k+2}$, let $sc_{k+1} = |c_{k+1} - p_{k+1}|$, $wc_{k+1} = sc_{k+1}/k$ and continue the processing using $X'$ as the new cut interval length with $p_{k+2}$ being the next anchor to be considered when placing next cut $c_{k+2}$. Otherwise, i.e., $D \geq U$, assume that the minimum weighted clearance $wc_{j'}(= U)$ occurs at $p_{j'}$. We know that if we select the cut interval length as $X_{wc} = X - wc_{j'}$, then cut $c_{j'}$ will be placed at $p_{j'}$, and there is exactly one anchor in each cut interval among $p_1$ to $p_{j'}$. The invariant condition (there is exactly one anchor in each cut interval) is maintained among $p_1$ to $p_{j'}$ if we use $X_{wc}$ to be the cut interval length. We can then separate the rest of anchors from $p_{j'}$ on to be a new subproblem such that the anchor $p_{j'}$ and cut $c_{j'}$ would play the role of $p_1$ and $c_1$ respectively, and the process repeats by using the cut interval length $X_{wc}$. Note that if there exists more than one weighted clearance $wc_{j'}$ which equals $U$, we pick the rightmost anchor $p_j$ with the largest index $j$.

In Case 2 we need at most $O(n)$ time to proceed to a cut interval containing more than one anchor, and recompute a new cut interval width, resulting in a new subproblem. Since there are at most $O(n)$ subproblems, the total time complexity is $O(n^2)$. We need $O(n)$ space to maintain the information of anchors, cuts and weighted clearances. We therefore conclude with the following theorem.

**Theorem 5.2** *The maximization version of* `1d-4S` *equal-width label placement problem can be solved in $O(n^2)$ time and $O(n)$ space.*

## 5.3 The Lower Bound

Now we consider the lower bound for the 1d-4S equal-width label placement problem. It is known that when the points are given sorted, the problem can be solved in linear time [3, 7]. We show that the problem requires $\Omega(n \log n)$ time, if the points are arbitrary. We adopt the algebraic computation model of Ben-Or [1] in which we have a random access machine with real arithmetic, and each arithmetic operation takes constant time. Consider the following decision version of the 1d-4S equal-width label placement problem.

**1d-4S equal-width label placement decision problem**

**Instance:** Given a set $P = \{p_1, p_2, \ldots, p_n\}$ of $n$ points on the real line and a width $w$.

**Output:** Yes, if there exists a disjoint set of $n$ intervals, each of width $w$ such that each interval contains exactly one point, and these intervals may have their endpoints common to each other, and no, otherwise.

We shall prove the lower bound of the above 1d-4S equal-width label decision problem by problem reduction [16]. Specifically we shall reduce to it the *uniform gap* problem, a well-known problem with an $\Omega(n \log n)$ lower bound, where $n$ is the input size [13]. The uniform gap problem is described in the following:

**Instance:** Given a set of $n$ real numbers $X = \{x_1, x_2, ..., x_n\}$ and a real number $\epsilon \geq 0$.

**Output:** Yes, if the gaps between consecutive numbers are uniformly equal to $\epsilon$ (Two numbers $x_i$ and $x_j$ are said to be consecutive if $x_i \leq x_j$ and there exists no other number $x_k$ such that $x_i < x_k < x_j$.); No, otherwise.

The problem reduction is as follows. Consider an instance $X = \{x_1, x_2, ..., x_n\}$. Assume $x_1$ is the minimum of $X$ and $x_n$ is the maximum. We must have $x_n - x_1 = (n-1)\epsilon$; else, return No obviously. Let $w = \epsilon$ and assign $p_1 = x_1$, $p_n = x_n - \epsilon$. We shall construct two equal-width label placement problem instances $I$ and $I'$. Let the instance $I$ with $p_i = x_i - \epsilon$, for each $p_i$, $1 < i < n$, and the instance $I'$ with $p_i = x_i$, for each $p_i$, $1 < i < n$. We claim that the gaps between consecutive numbers of $X$ are uniformly equal to $\epsilon$ if and only if 1d-4S equal-width label placement decision problem has feasible solutions for both instances $I$ and $I'$. If the gaps between consecutive numbers of $X$ are uniformly equal to $\epsilon$, then for instance $I$ the second smallest point is also located in $p_1$ and the other points excluding $p_1$ and $p_n$ are located in uniform distribution of length $(n-2)\epsilon$ by interleaving $\epsilon$-intervals between $p_1$ and $p_n$, in the instance $I$. Therefore, except that the label of $p_1$ is placed to the left of $p_1$, the labels of all the rest of points are placed to the right of their anchors. On the other hand, for instance $I'$, the second largest point is located in $p_n$ and the other points excluding $p_1$ and $p_n$ are located in uniform distribution of length $(n-2)\epsilon$ by interleaving $\epsilon$-intervals between $p_1$ and $p_n$. Similarly, except that the label of $p_n$ is placed to the right of $p_n$, the labels of the rest of points are placed to the left of their anchors. It is obvious that there exist feasible solutions for both instances $I$ and $I'$. Conversely, suppose there exist consecutive numbers $x_i$ and $x_j$ with $|x_i - x_j| \neq \epsilon$ in $X$. Assume $x_j > x_i$ without loss of generality. It divides into two cases. If $x_j - x_i < \epsilon$, then $p_j = x_j - \epsilon < x_i = (x_i - \epsilon) + \epsilon = p_i + w$ for instance $I$. Since the only feasible label placement in instance $I$ is that each label is placed to the right of its anchor, there is no feasible solution for instance $I$. The other case is $x_j - x_i > \epsilon$. It implies that $p_j - w > p_i$ for instance $I'$. Since the only feasible label placement in instance $I'$ is that each label is placed to the left of its anchor, there exists $p_k$ between $p_j$ and $p_n$ such that its label does not have sufficient space $\epsilon$. Therefore there is no feasible solution for instance $I'$. This completes the proof.

Thus from the above we obtain the following theorem.

**Theorem 5.3** *The* `1d-4S` *equal-width label placement problem for $n$ points on a line requires $\Omega(n \log n)$ time, under the algebraic computation tree model.*

**Corollary 5.4** *The* `1d-4S` *equal-width label placement problem for $n$ points on a line can be solved in $O(n \log n)$ time, which is asymptotically optimal.*

**Proof.** This follows from the above theorem and the results of [3], [7] that the 1d-4S equal-width label placement problem can be solved in linear time when the anchors are sorted. □

# 6 Conclusion

In this paper we have presented a linear time algorithm for solving Slope-4P unit-height (width) label placement problem provided that the input points are given sorted. We have also proved that the Slope-4P elastic label placement problem and the Slope-4P unit-height (width) label placement problem are equivalent. We have considered an optimization (maximization) version of 1d-4S equal-width label placement problem and provided a new method to solve it in $O(n^2)$ time. In addition, we have presented a lower bound proof for 1d-4S equal-width label placement problem. The above algorithms have been implemented using GeoBuilder and the details can be found at http://www.sharetone.org/Members/jhh/GeoBuilder/system_demo.htm.

Although Slope-4P elastic label and Slope-4P unit-height (width) label are equivalent, how to choose an appropriate height and width for each label to make these labels look aesthetically nice remains an open problem. We also feel that the complexity results of optimization (maximization) version could be improved. Whether there exist solutions for the Slope-4P arbitrary rectangle label model remains to be seen.

**Acknowledgements.** We would like to thank the anonymous referees for their comments on an earlier version that have helped improve the presentation. The earlier version was based on the Master thesis of Yu-Shin Chen [3], Dept. of Computer Science and Information Engineering, National Taiwan Univ., June 2003.

# References

[1] M. Ben-Or, Lower bounds for algebraic computation trees, *In Proc. 15th Annu. ACM Sympos. Theory Comput.*, 1983, pp. 80-86.

[2] B. Chazelle and 36 co-authors. The computational geometry impact task force report. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223, pages 407-463. American Mathematical Society, Providence, 1999.

[3] Yu-Shin Chen, Labeling points on a single line, Master Thesis, Dept. Computer Science and Information Engineering, National Taiwan Univ., June 2003.

[4] J. Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point feature label placement. *ACM Transactions on Graphics*, 14(3), pp. 203-232, 1995.

[5] R. Duncan, J. Qian, A. Vigneron, and B. Zhu, Polynomial time algorithms for three-label point labeling, *Theoretical Computer Science*, 296(1), pp. 75-87, 2003.

[6] M. Formann and F. Wagner. A packing problem with applications in lettering of maps. *Proceedings of the 7th ACM Symposium on Computational Geometry.* (1991) 281-288.

[7] M. Á Garrido, C. Iturriaga, A. Márquez, J. R. Portillo, P. Reyes, and A. Wolff. Labeling Subway Lines. *Proc. 12th Annual International Symposium on Algorithms and Computation (ISAAC'01)*, volume 2223 of Lecture Notes in Computer Science, pp. 649-659.

[8] C. Iturriaga and A. Lubiw. Elastic labels: The two-axis case. *Proceedings of the Symposium on Graph Drawing (GD'97)*, volume 1353 of Lecture Notes in Computer Science, pp. 181-192, 1997.

[9] Claudia Iturriaga and Anna Lubiw. Elastic labels around the perimeter of a map. *Journal of Algorithms*, 47(1), pp. 14-39, 2003.

[10] T. Kato and H.Imai. The NP-completeness of the character placement problem of 2 or 3 degrees of freedom. *In Record of Joint Conference of Electrical and Electronic engineers in Kyushu.* page 1138, 1988. In Japanese.

[11] D. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM Disc. Math.* 5(3), pp. 422-427, 1992.

[12] M. van Kreveld, T. Strijk, and A. Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13:21-47, 1999.

[13] D. T. Lee and Y. F. Wu, Geometric complexity of some location problems, *Algorithmica*, 1(1986), pp. 193-211.

[14] J. Marks and S. Shieber. The computational complexity of cartographic label placement. *Technical Report TR-05-91.* Harvard University CS (1991).

[15] S.-H. Poon, C.-S. Shin, T. Strijk, T. Uno, and A. Wolff. Labeling points with weights. *Algorithmica*, 38(2):341-362, 2003.

[16] F. P. Preparata and M. I. Shamos, Computational Geometry : An Introduction, Springer-Verlag, 1993.

[17] T. Strijk and A. Wolff . Labeling points with circles. *International Journal of Computational Geometry and Applications (IJCGA)* 11(2):181–195, April 2001.

[18] A. Wolff and T. Strijk. The Map-Labeling Bibliography. http://i11www.ira.uka.de/map-labeling/bibliography/, 1996.

[19] F. Wagner and A. Wolff. A practical map labeling algorithm. *Computational Geometry: Theory and Applications.* 7:387-404, 1997.